Review Test Submission: ECE 344L Test 2 Spring 2020

User	David Kirby
Course	ECE-344L-003 (Spring 2020)
Test	ECE 344L Test 2 Spring 2020
Started	4/21/20 8:57 PM
Submitted	4/22/20 3:31 PM
Due Date	4/24/20 12:00 PM
Status	Completed
Attempt Score	102 out of 100 points
Time Elapsed	18 hours, 33 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

4 out of 4 points (Extra Credit)

Which of the following is NOT a serial communication interface present on the MX7 board?

Selected Answer: ✓ RS232

Answers: ✓ RS232

UART

SPI

I2C

Question 2

3 out of 3 points (Extra Credit)

Which PIC32 microcontroller is included with the DigilentPro MX7?

Selected Answer: ✓ B. PIC32MX795F512L

Answers: A. PIC32MX795F512H

B. PIC32MX795F512L

C. PIC32MX460F512L

D. PIC32MX460F512H

Question 3

12 out of 12 points

Describe the characteristics of a half-duplex, synchronous, serial communication channel. Also, identify the minimal set of signal lines that will be needed to connect communicating devices which are configured with an interface with these characteristics.

Selected Answer:

Half-duplex: only allows data to be received or transmitted at a time, not both simultaneously.
Synchronous: one of the devices, typically the master, transmits a clock and all other devices (slaves) are receiving.
Serial communication: only transmits one bit at a time, however recent advances have increased the speed of this communication considerably.
Half-duplex, synchronous, serial communication describes I2C which only uses two wires - the serial data line and the serial clock line.

Correct Answer:

Serial: one bit at a time
Half Duplex: Data flows in only one direction at a time. Tx or Rx
Synchronous: A master device transmits a clock
Signals needed: clock and data (similar to I2C - SCK and SDA)

Question 4

12 out of 12 points

TABLE 12-1: PORTA REGISTER MAP FOR PIC32MX534F064L, PIC32MX564F064L, PIC32MX564F128L, PIC32MX575F256L, PIC32MX575F512L, PIC32MX664F064L, PIC32MX664F128L, PIC32MX675F256L, PIC32MX675F512L, PIC32MX695F512L, PIC32MX764F128L, PIC32MX775F256L, PIC32MX775F512L AND PIC32MX795F512L DEVICES

Virtual Address (BF88_#)	Register Name(1)	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
6000	TRISA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	TRISA15	TRISA14	—	—	—	TRISA10	TRISA9	—	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	C6FF
6010	PORTA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	RA15	RA14	—	—	—	RA10	RA9	—	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
6020	LATA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	LATA15	LATA14	—	—	—	LATA10	LATA9	—	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
6030	ODCA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	ODCA15	ODCA14	—	—	—	ODCA10	ODCA9	—	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	0000

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, resctively. See Section 12.1.1 “CLR, SET and INV Registers” for more information.

The LAT register is the register used to write data to the port I/O pins. Assume we have configured all of the Port A pins as outputs, and each is pin is currently set to a desired value. Now, we need to clear bit 9 (set to zero) while not affecting any of the other bits. What is the hexadecimal value and the address that we must write to, in order to set the specific bit to zero

while not affecting any of the other bits?

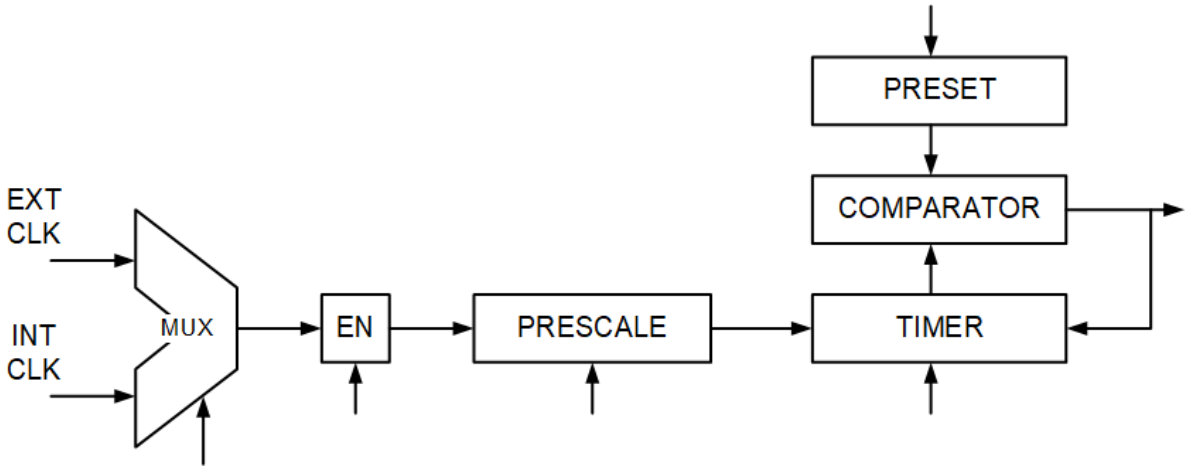
Selected Answer: `Write the value 0x200 to the address 0xBF886024.`

Correct Answer: Address: 0xBF886024
Hex Value: 0x200

Question 5

10 out of 10 points

You are using a 16-bit timer, timer 1, which has pre-scale options of 1:1, 1:8, 1:64, and 1:256. You are using PBCLK, an internal clock, which is configured to run at 10MHz.



Is it possible to set the preset register to a value that corresponds to a time interval of 1.70 seconds, so that the timer can count from 0 to the value and generate an interrupt? You must justify your answer. A yes/no answer will get no points.

Selected Answer: `Since this is a 16-bit timer, the limit is 65535. Using the maximum prescaler (1:256) we would get 10MHz/256 = 39.063kHz. At a time interval of 1.7 seconds, this would result in an operation of 1.7s x 39.063kHz = 66406 which is greater than the 16-bit limit. Therefore, no it is not possible to set the preset register to a value corresponding to 1.70 seconds.`

Correct Answer:
10Mhz/256 = 39,062 Hz or T = 25.6 uSec

25.6uSec/count * 65,535 counts = 1.677 Sec
Therefore, No, we can not measure a time interval of 1.70 sec using a 16 bit timer.

Question 6

5 out of 5 points

Which peripheral on the PIC32MX795F512L would we use if we need to implement a full-

duplex, synchronous, serial communication channel?

Selected Answer: SPI uses full-duplex, synchronous, serial communication.
Correct Answer: SPI

Question 7

5 out of 5 points

In the PIC microcontroller, we use SFRs. What are SFRs used for and how do they differ from general purpose registers?

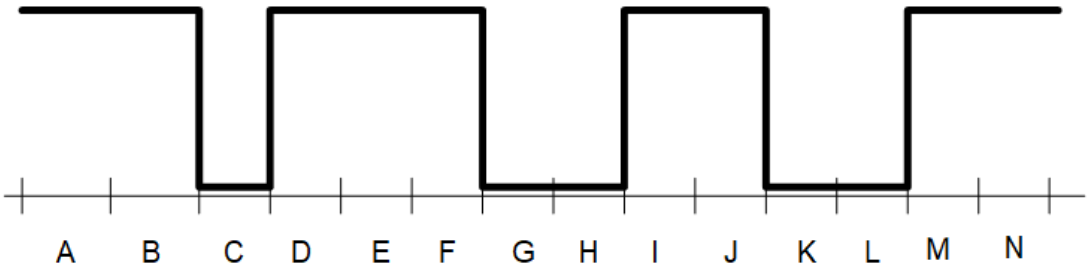
Selected Answer: SFRs are special function registers that are used to configure and control hardware and peripheral functions. General purpose registers are used for data storage and address calculations only, no instructions.

Correct Answer: SFRs are used to: configure hardware, configure peripherals, communicate with peripherals, and monitor status.
General purpose registers are used to strore data and addresses.

Question 8

6 out of 14 points

You have configured the UART on your PIC32 microcontroller to communicate at 2400 baud using an 8,N,1 configuration, where the X,Y,Z notation corresponds to data bits, parity, and stop bits. The terminal to which you are connected is transmitting signals to your microcontroller, but the data are not received properly so you instrument the connection with the oscilloscope and observe the following signal, where each tick on the line corresponds to one bit time:



You measure the bit time and it corresponds to a 2400 baud rate, but the data being received erroneously. Analyze the signal and identify each bit type using the options below to give you insight into what the sender configuration might be. Match the bit times indicated by the letters with the indicated bit types.

Question	Correct Match	Selected Match

A	✓ 1. Idlel bit	✓ 1. Idlel bit
B	✓ 1. Idlel bit	✓ 1. Idlel bit
C	✓ 2. Start Bit	✓ 2. Start Bit
D	✓ 3. Data Bit	✗ 1. Idlel bit
E	✓ 3. Data Bit	✗ 1. Idlel bit
F	✓ 3. Data Bit	✗ 1. Idlel bit
G	✓ 3. Data Bit	✓ 3. Data Bit
H	✓ 3. Data Bit	✓ 3. Data Bit
I	✓ 3. Data Bit	✗ 1. Idlel bit
J	✓ 3. Data Bit	✗ 1. Idlel bit
K	✓ 3. Data Bit	✓ 3. Data Bit
L	✓ 4. Data or Parity Bit	✗ 5. Stop Bit
M	✓ 5. Stop Bit	✗ 1. Idlel bit
N	✓ 6. Idle or Stop Bit	✗ 1. Idlel bit

All Answer Choices

1. Idlel bit
2. Start Bit
3. Data Bit
4. Data or Parity Bit
5. Stop Bit
6. Idle or Stop Bit

Question 9

10 out of 10 points

You are examining a byte of digital data received from the SPI interface. You are processing the received data as ASCII characters. The byte read from the SPI buffer has the following value: 10010011. Is this a valid ASCII character? Why, or why not?

Selected Answer: The ASCII standard is 7 bits with a range of 0:127. The value 1001 0011 = 179 in decimal and is therefore out of range and not a valid ASCII character. Also, we could have looked at the most significant bit, seen that it was a one and determined that this was not a valid ASCII character.

Correct Answer: No, the msb is not 0. This is not a valid ASCII character.

Question 10 10 out of 10 points

What are the three basic requirements that must be met to use interrupts on the PIC microcontroller?

Selected Answer:
1. Need an interrupt source
2. Enable specific interrupt
3. Need an ISR (interrupt service routine)

Correct Answer:
1. Need an interrupt source
2. Must enable the specific interrupt
3. Need and Interrupt Service Routine (ISR)

Question 11 10 out of 10 points

You are debugging a PIC32 program and are examining values of 32 bit operands in memory. You examine one location and the value is 0x80000010. What are the **two** decimal equivalents of this hexadecimal value if the operand is an unsigned integer or a signed integer?

(Hint: 2^{31} = 2,147,483,648)

Selected Answer:
0x80000010 = 1000 0000 0000 0000 0000 0000 0001 0000 (binary)
= 2,147,483,664 (unsigned integer)
= -2,147,483,648 + 16 = -2,147,483,632 (signed integer)

Correct Answer:
Unsigned: 2,147,483,648 + 16 = 2,147,483,664

Signed: -2,127,483,648 + 16 = - 2,147,483,632

Question 12 3 out of 3 points (Extra Credit)

If you wanted to use the input: BTN1, which Port and Pin would you need?

Selected Answer: RG6

C.

Answers:
A. RA7
B. RG7
C. RG6
D. RA10

Question 13

12 out of 12 points

TABLE 7-7: INTERRUPT REGISTER MAP FOR PIC32MX764F128L, PIC32MX775F256L, PIC32MX775F512L AND PIC32MX795F512L DEVICES

Virtual Address (BF88_#)	Register Name(1)	Bit Range	Bits 31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Resets
1000	INTCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SS0	0000
		15:0	—	—	—	MVEC	—	TPC<2:0>			—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000
1010	INTSTAT(3)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	SRIPL<2:0>			—	—	VEC<5:0>						0000
1020	IPTMR	31:16	IPTMR<31:0>																0000
		15:0																	0000
1030	IFS0	31:16	I2C1MIF	I2C1SIF	I2C1BIF	U1TXIF SPI3TXIF I2C3MIF	U1RXIF SPI3RXIF I2C3SIF	U1EIF SPI3EIF I2C3BIF	SPI1TXIF	SPI1RXIF	SPI1EIF	OC5IF	IC5IF	T5IF	INT4IF	OC4IF	IC4IF	T4IF	0000
		15:0	INT3IF	OC3IF	IC3IF	T3IF	INT2IF	OC2IF	IC2IF	T2IF	INT1IF	OC1IF	IC1IF	T1IF	INT0IF	CS1IF	CS0IF	CTIF	0000
1040	IFS1	31:16	IC3EIF	IC2EIF	IC1EIF	ETHIF	CAN2IF(2)	CAN1IF	USBIF	FCEIF	DMA7IF(2)	DMA6IF(2)	DMA5IF(2)	DMA4IF(2)	DMA3IF	DMA2IF	DMA1IF	DMA0IF	0000
		15:0	RTCCIF	FSCMIF	I2C2MIF	I2C2SIF	I2C2BIF	U2TXIF SPI4TXIF I2C5MIF	U2RXIF SPI4RXIF I2C5SIF	U2EIF SPI4EIF I2C5BIF	U3TXIF SPI2TXIF I2C4MIF	U3RXIF SPI2RXIF I2C4SIF	U3EIF SPI2EIF I2C4BIF	CMP2IF	CMP1IF	PMPIF	AD1IF	CNIF	0000
1050	IFS2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	U5TXIF	U5RXIF	U5EIF	U6TXIF	U6RXIF	U6EIF	U4TXIF	U4RXIF	U4EIF	PMPEIF	IC5EIF	IC4EIF	0000
1060	IEC0	31:16	I2C1MIE	I2C1SIE	I2C1BIE	U1TXIE SPI3TXIE I2C3MIE	U1RXIE SPI3RXIE I2C3SIE	U1EIE SPI3EIE I2C3BIE	SPI1TXIE	SPI1RXIE	SPI1EIE	OC5IE	IC5IE	T5IE	INT4IE	OC4IE	IC4IE	T4IE	0000
		15:0	INT3IE	OC3IE	IC3IE	T3IE	INT2IE	OC2IE	IC2IE	T2IE	INT1IE	OC1IE	IC1IE	T1IE	INT0IE	CS1IE	CS0IE	CTIE	0000
1070	IEC1	31:16	IC3EIE	IC2EIE	IC1EIE	ETHIE	CAN2IE(2)	CAN1IE	USBIE	FCEIE	DMA7IE(2)	DMA6IE(2)	DMA5IE(2)	DMA4IE(2)	DMA3IE	DMA2IE	DMA1IE	DMA0IE	0000
		15:0	RTCCIE	FSCMIE	I2C2MIE	I2C2SIE	I2C2BIE	U2TXIE SPI4TXIE I2C5MIE	U2RXIE SPI4RXIE I2C5SIE	U2EIE SPI4EIE I2C5BIE	U3TXIE SPI2TXIE I2C4MIE	U3RXIE SPI2RXIE I2C4SIE	U3EIE SPI2EIE I2C4BIE	CMP2IE	CMP1IE	PMPIE	AD1IE	CNIE	0000
1080	IEC2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	U5TXIE	U5RXIE	U5EIE	U6TXIE	U6RXIE	U6EIE	U4TXIE	U4RXIE	U4EIE	PMPEIE	IC5EIE	IC4EIE	0000
1090	IPC0	31:16	—	—	—	INT0IP<2:0>			INT0IS<1:0>		—	—	—	CS1IP<2:0>			CS1IS<1:0>		0000
		15:0	—	—	—	CS0IP<2:0>			CS0IS<1:0>		—	—	—	CTIP<2:0>			CTIS<1:0>		0000
10A0	IPC1	31:16	—	—	—	INT1IP<2:0>			INT1IS<1:0>		—	—	—	OC1IP<2:0>			OC1IS<1:0>		0000
		15:0	—	—	—	IC1IP<2:0>			IC1IS<1:0>		—	—	—	T1IP<2:0>			T1IS<1:0>		0000
10B0	IPC2	31:16	—	—	—	INT2IP<2:0>			INT2IS<1:0>		—	—	—	OC2IP<2:0>			OC2IS<1:0>		0000
		15:0	—	—	—	IC2IP<2:0>			IC2IS<1:0>		—	—	—	T2IP<2:0>			T2IS<1:0>		0000
10C0	IPC3	31:16	—	—	—	INT3IP<2:0>			INT3IS<1:0>		—	—	—	OC3IP<2:0>			OC3IS<1:0>		0000
		15:0	—	—	—	IC3IP<2:0>			IC3IS<1:0>		—	—	—	T3IP<2:0>			T3IS<1:0>		0000

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Except where noted, all registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See Section 12.1.1 "CLR, SET and INV Registers" for more information.
2: This bit is unimplemented on PIC32MX764F128L device.
3: This register does not have associated CLR, SET, and INV registers.

Using table 7.7 above specify the address and the 32-bit word (expressed as a hexadecimal values) that must be written to it, in order to enable the interrupt for Timer 2

Selected Answer: `Write the value 0x100 to the address 0xBF881060.`

Correct Answer:
bit 8 of IEC0 SFR
Address: 0xBF881060
Hex Value: 0x100

Monday, May 4, 2020 11:27:50 AM MDT

← OK